DOCUMENT RESUME

ED 479 619 IR 022 223

AUTHOR	Tappert, Charles C.
TITLE	Students Develop Real-World Web and Pervasive Computing Systems.
PUB DATE	2002-10-00
NOTE	10p.; In: E-Learn 2002 World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education. Proceedings (7th, Montreal, Quebec, Canada, October 15-19, 2002); see IR 022 093.
AVAILABLE FROM	Association for the Advancement of Computing in Education (AACE), P.O. Box 3728, Norfolk, VA 23514. Tel: 757-623-7588; e-mail: info@aace.org; Web site: http://www.aace.org/DL/.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	EDRS Price MF01 Plus Postage. PC Not Available from EDRS
DESCRIPTORS	*Computer Science Education; *Computer System Design; Cooperative Learning; Experiential Learning; Higher Education; Information Technology; Instructional Effectiveness; *Student Developed Materials; *Student Projects; Teamwork
IDENTIFIERS	Pace University NY

ABSTRACT

In the academic year 2001-2002, Pace University (New York) Computer Science and Information Systems (CSIS) students developed real-world Web and pervasive computing systems for actual customers. This paper describes the general use of team projects in CSIS at Pace University, the real-world projects from this academic year, the benefits of including real-world projects in the curriculum, and their cost in terms of the instructor load and the project development infrastructure. The benefits of the method of structuring and conducting the real-world projects were found to far outweigh the cost of providing them. The projects provide valuable systems for the customers, allow the students to develop technical and value skills, foster interdisciplinary collaboration, encourage student involvement in the university and local communities, support student and faculty research, enhance relationships between the university and local technology companies, and increase national recognition of the university. (Contains 23 references.) (Author)

Reproductions supplied by EDRS are the best that can be made
from the original document.

ED 479 619

Students Develop Real-World Web and Pervasive Computing Systems

By: Charles C. Tappert

PERMISSION TO REPRODUCE AND
DISSEMINATE THIS MATERIAL IN
MICROFICHE ONLY HAS BEEN GRANTED BY

G.H. Marks

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)

2B

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

IR022223

BEST COPY AVAILABLE

Students Develop Real-World Web and Pervasive Computing Systems

Charles C. Tappert
School of Computer Science and Information Systems
Pace University, USA, ctappert@pace.edu

Abstract: In the academic year 2001-2, Pace University CSIS students developed real-world Web and pervasive computing systems for actual customers. We describe the general use of team projects in CSIS at Pace University, the real-world projects from this academic year, the benefits of including real-world projects in the curriculum, and their cost in terms of the instructor load and the project development infrastructure. We found that the benefits of our method of structuring and conducting the real-world projects far outweigh the cost of providing them. The projects provide valuable systems for the customers, allow the students to develop technical and value skills, foster interdisciplinary collaboration, encourage student involvement in the university and local communities, support student and faculty research, enhance relationships between the university and local technology companies, and increase national recognition of the university.

Introduction

In the academic year 2001-2, the School of Computer Science and Information Systems (CSIS) at Pace University used real-world Web and pervasive computing systems as team projects in a two-course capstone sequence in Software Engineering, and in an elective course in Pervasive Computing Systems, in the Master of Science in Computer Science program. Although the students in both these courses develop systems for actual customers, the emphasis of the project work in these two courses is somewhat different. In the software engineering sequence, the customers are referred to as clients. The clients are the source of the project requirements, and the teams focus on developing systems that meet the real needs of the clients. In the pervasive computing course, student invention teams, called E-Teams, the "E" standing for "entrepreneurship," work with customers to build innovative systems that use pervasive computing techniques and that demonstrate the feasibility of potential commercial products. In that course, the customers are referred to as mentors since they are typically the knowledge sources for the systems. In this paper, the term "customer" will be used for both client and mentor in describing 18 student projects.

Developing real-world systems for real customers provides an excellent setting for teaching students team and value skills by following a human-centered development process. While it is clear that students of computing must learn technical skills, current thinking is that they must also learn "value skills" of how to interact with customers to provide them with the promised value-producing systems. Six categories of value skills are described in (Denning & Dunham 2001): coordination (of requests, offers, promises, etc.); customer relations; commitment management; teamwork; lifelong learning; and business and entrepreneurship skills. Dertouzos (2001) also makes a strong case for human-centered computing in which technology adapts to people rather than the other way around.

The real-world projects provide valuable systems for the customers, allow the students to develop technical and value skills, utilize student-centered learning, foster interdisciplinary collaboration, encourage student involvement in the university and local communities, support student and faculty research, enhance relationships between the university and local technology companies, and increase national recognition of the university.

The following sections describe the general use of team projects in CSIS at Pace University, the real-world projects from the 2001-2 academic year, the benefits of including real-world projects in the CSIS curriculum, their cost in terms of the instructor load and the project development infrastructure, and conclusions.

Team Projects in CSIS at Pace University

The Computer Science department at Pace University has frequently used team projects to provide students with the educational experience of working together in teams, similar to what is done in industry, in order to design, build, and test computer information systems. For several years prior to 2001-2 the software engineering course sequence had each student team develop as its project an "Order Processing System." For these "canned" projects the instructor has played a dual role, acting as both project manager and customer.

A project team typically consists of 4 to 5 students: an Architect-Designer, one or two Implementers, a Quality Officer, and a team Coordinator-Liaison (Pressman 2001). Ideally, teams should meet once a week in addition to project work time. Team members should work together and coordinate activities as a unit – for example, communication, including e-mail, between the team and the customer should be done as a group or should be coordinated by the team leader. In recent years, Java has been the preferred language at Pace University for projects

that require programming. Although the requirements for the projects come from the customers, the course instructor is the "boss," or "Chief Executive Officer," of the project teams – that is, the person who makes all the major decisions and adjudicates personnel problems.

Real-World Projects in 2001-2 in CSIS at Pace University

Using real-world projects with real customers is not new; for example, the University of Southern California's Center for Software Engineering (USC CSE Website 2002) has been doing this for years in their capstone two-semester software engineering sequence. However, we believe that our program is unique because it focuses on value skills, serves the community, and is organized on a large scale with supporting project infrastructure.

Eighteen projects were completed during the 2001-2 academic year (Table 1). Ten were Web interfaces to backend databases, four were medical applications, three concerned VoiceXML or InkXML, and one was a cluster and grid computing system. The customers were from various Pace University schools/departments and from several outside organizations. The university schools/departments included the School of CSIS; the Lienhard School of Nursing; the Lubin School of Business; the Office of Planning, Assessment, Research, and Academic Support; the Department of Information Technology (DoIT); and the Doctor of Professional Studies (DPS) in Computing program. The outside organizations included Northern Westchester Hospital, IBM Research, the Psychology Department at SUNY New Paltz, and the Fulbright scholarship program. The table column on scripting/database software will be discussed later in the section on project development infrastructure.

	Project	Customer Affiliation	Scripting/database software
1	Online Course Survey System	Planning & Assessment/Pace	ASP/Microsoft Access
2	Project Group Assignment System	CSIS/Pace	Cold Fusion/Microsoft Access
3	Test Item Reliability Analyzer	CSIS/Pace	PHP/MySQL
4	Genealogy Web Application	CSIS/Pace	Cold Fusion /MySQL
5	Dean's Communication System	CSIS/Pace	Cold Fusion/Microsoft Access
6	PC Maintenance/Tracking System	CSIS/Pace	Cold Fusion/Microsoft Access
7	DPS External Website	DPS Program/Pace	PHP/MySQL
8	Experimental Handwriting System	CSIS/Pace	ASP/Microsoft Access
9	Rare Coin Grading System	DPS Program/Pace	ASP/Microsoft Access
10	User Verification System	CSIS/Pace	ASP/Microsoft Access
11	Nurse Information System (NIS)	Nursing/Pace and DoIT/Pace	–
12	NIS Wireless Extension	Nursing/Pace and DoIT/Pace	–
13	Emergency Pre-Hospital Care Communication System	Northern Westchester Hospital and Business/Pace	–
14	Medical Vital Sign Wearable	CSIS/Pace and IBM Research	–
15	VoiceXML App. Devel. Facility	CSIS/Pace and IBM Research	–
16	VoiceXML Applications	DoIT/Pace, CSIS/Pace, and Psychology/SUNY New Paltz	PHP/MySQL
17	Multimodal Voice/InkXML System	Fulbright Program and IBM	–
18	Cluster/Grid Computing System	CSIS/Pace and DoIT/Pace	–

Table 1: The eighteen Web and pervasive computing projects in 2001-2.

The student teams were required to write technical papers for possible publication describing their systems. Eight, referenced herein, were presented at this year's Mid-Atlantic Student Workshop on Programming Languages and Systems, and PDF versions are available from the workshop's Website (MASPLAS'02 Website 2002).

Web interface to backend database client/server systems

Ten of the eighteen systems consisted of a Web interface to a backend database using the client/server architecture. The simpler ones simply allowed users to enter information into the database and retrieved the data for appropriate viewing. The more complex ones performed calculations on the data, which in some cases was substantial.

The **Online Course Survey System** (Pethe, et al. 2002) was designed to automate Pace University's manual course-end survey system, in which students complete paper surveys to evaluate the courses they have taken. With the new system, students log in to take the survey through a Web interface, results are retained in a database, and survey statistics can be accessed from the Website. The system was tested by running in parallel both the old and the new survey systems on two large CSIS courses at the end of the spring semester. It was also used to obtain surveys for all online CSIS courses, courses for which it was previously not possible to obtain survey results, since

BEST COPY AVAILABLE

the students in these classes do not physically attend class and therefore cannot be given a paper survey to fill out. These tests were successful and it is anticipated that the whole university will soon move to an online system.

The **Project Group Assignment System** (Doyle et al. 2002) is a Web-based system that creates teams (groups) for project work in industry or academia. This system addresses problems that arise in the usual, manual team-creation process, which can be tedious, time consuming, perceived as unfair, and may not get a good mix of expertise. In the automated system, the project managers enter the project information: project descriptions, suggested team size; customer information; and parameter weights for team candidates' project preferences, availability, geography, and experience. Potential team members complete an online survey to record their project preferences, preferred availability times and locations for team meetings, and their academic and/or professional experience. The project manager then executes a Java program that groups the candidates into teams, and team members and program managers can view the resulting teams through the Web interface. Since an exhaustive search for the optimal solution is not possible for more than a small number of projects, the algorithm uses a greedy search approach to obtain a near-optimal solution similar to a previous stand-alone program (Redmond 2001).

The **Test Item Reliability Analyzer** allows instructors to perform statistical analysis on multiple-answer tests. The measures provided by the system give the instructor greater insight into the relative preparedness of the students as well as the validity of individual test questions. The system computes five item (test question) measures: item difficulty (the percentage of students answering the item correctly), item variance (a measure of the variability of the student responses), item discrimination (a measure of the discrimination power of a test item), item correlation (a measure of the correlation between the item and the whole test), and item distraction (incorrect alternatives for a test item that are never chosen and should therefore be removed or replaced). It also computes an aggregate analysis indicator of overall test quality and accuracy. Such objective measures of the questions are valuable, since the things that separate good items from bad items are best not left to the subjective interpretation of the test constructor. The one similar system found (ITEMAL 2002) is proprietary to the University of Delaware, was never intended for public distribution, and is not very flexible or user-friendly. The new system successfully analyzed the quizzes used in the software engineering course and indicated where improvements can be made.

The **Genealogy Web Application** (Barbosa et al. 2002) is an example of developing a Web-based application that improves on an existing commercial product – the shrink-wrapped, stand-alone software application packages for the Microsoft Windows Operating System environment. With the stand-alone genealogy systems, usually one family member researches, inputs, and publishes his family tree and thereby undertake the tedious and never-ending task of updating and maintaining it, a particularly burdensome task for large family trees. In contrast, the Web-based application distributes this burden by allowing several family members to update and maintain their branch of the family tree and to do so remotely from anywhere through a Web interface. Furthermore, all family members can view the family tree information on the Web.

The **Dean's Communication System** assists the Dean of the School of CSIS in her communications with faculty, students, and others at Pace University. It consists of a database backend with both an external and an internal Web interface. The external "Communications from the Dean" site is accessible by all, and the internal one is accessible only by those having access permission. All communication items are stored in the database for tracking and for historical record keeping.

The **PC Maintenance/Tracking System** assists in the maintenance, support, and tracking of PCs in CSIS at Pace University. Information is captured when a PC is reported as having a problem or needing an upgrade (information such as the date, the PC ID, the PC's location, the nature of problem or upgrade), when the problem is fixed (what repairs were made), when software is installed (the software, the installer), when a PC is moved, etc.

The **DPS External Website** team developed a new Website for the Doctor of Professional Studies (DPS) in Computing program (Merritt et al. 2001). This site improves on the old site in terms of elegance, clarity, and ease of navigation. It has a backend database and a link to a back office interface that allows administrators without knowledge of HTML to change or update the look and content of the Website. The site also allows visitors to request information and to submit an application online.

An **Experimental Handwriting System** was developed to investigate the detection of handwriting forgery. The team collected handwriting samples from ten subjects who wrote a set of words in their natural style and attempted forgeries of the other nine subjects' handwriting. These handwriting samples were digitally scanned and stored in an image database. A Web-based quiz was developed to gather statistical data from users trying to distinguish between the original and the forged handwriting samples. Novices and certified document examiners (CDEs) were asked to take the online quiz. Although many students took the quiz as novices, only a few CDEs were willing to do so, and the results were inconclusive. The implications of this and similar studies may have bearing on the admissibility of handwriting testimony in U.S. courts. The database of handwriting samples was also used in a study of handwriting forgery detection by machine (Cha & Tappert 2002).

The **Rare Coin Grading System** (Bassett et al. 2002) was designed to automate the grading of rare coins. In the numismatic community today, the official grading of coins is accomplished by human inspection. This, by its very nature, lends itself to inconsistent results that can misrepresent the actual value of a coin by thousands of dollars; there is also the danger of misrepresentation for personal gain. The team obtained graded coins and created a database for storing their images. Then, they developed a Web interface to an educational quiz environment in which experts and novices can try their hand at grading the coins in the database. Finally, they developed a rudimentary machine grader by extracting three histograms (hue, saturation, and brightness) and an edge detection function from the digitally scanned color image of the coin to be graded, and assigning that coin the grade of the pre-graded coin that it most closely matched. Preliminary results of the machine process appear reasonable and indicate the promise of the approach.

The **User Verification System** (Baker et al. 2002) was developed to investigate combining several biometrics in order to develop a high-confidence user verification system. Such a subsystem can be embedded into any system that requires high security and can be developed using various combinations of different modalities; this project used four: face, fingerprint, handwriting, and voice. Each of the four verification systems involves techniques of pattern recognition to compare signal samples from the input device with those from the database, and each can result in type I (valid user rejected) and type II (invalid user accepted) errors. Combining these four different verification techniques should achieve a high level of confidence. Initial work on this problem focused primarily on face recognition and resulted in a database of digitized photos and preliminary studies of verification by face image.

Medical application systems

Four systems involved medical applications. The **Nurse Information System (NIS)** (Palmer et al. 2002) physical assessment application walks a student nurse through a physical assessment. A legacy system on a proprietary device running an obsolete operating system was ported to a Java 2 Micro Edition (J2ME) implementation. The application was completely redesigned and rewritten using object-oriented design techniques to create portable classes for reuse and for use in future applications. A small prototype was built and scaled up to a full application on Palm personal digital assistants (PDAs) running the PalmOS. A C++ conduit was also written to allow the prototype to transfer data from the Palm handheld devices to a PC for storage and further evaluation.

The same students who developed the above-described NIS in the software engineering course extended that system with an **NIS Wireless Extension** in the pervasive computing course. They provided the earlier system with a wireless extension through the use of Java servlets on a Web server to allow the data to be e-mailed as an attachment to a user-specified e-mail address (Palmer et al. 2002). This ensures that the NIS has a cross-platform future and can migrate from one device to another to take full advantage of the J2ME networking features.

A preliminary study of an **Emergency Pre-Hospital Care Communication System** was conducted for Northern Westchester Hospital (Park et al.). Many hospital emergency departments are exploring new communication and information technologies that will assist them in providing higher quality of care by improving the speed of flow, the consistency, and the accuracy of information shared among all the parties involved in an emergency response team. This study describes some currently available or on-the-horizon communication and information technologies that may be appropriate for use in the emergency services field, discusses three systems currently being used by other hospitals, and provides three alternative approaches to obtaining the type of emergency communication system that will best fit the needs of Northern Westchester Hospital's emergency department. The study found that the solution to be chosen should depend on the time frame allotted for the project, the funds budgeted, and the number and skills of the staff members responsible for implementing the system.

The **Medical Vital Sign Wearable** computer monitors a patient's vital signs, specifically heart rate and breath rate, which serve as indicators of a patient's immediate general health. If performed on a continuous basis and monitored remotely, it could assist medical support personnel. For example, each patient could be equipped with a medical status monitor that constantly reads blood pressure, pulse, respiration, and blood oxygen level, and a built-in expert system could determine when the patient's condition warrants attention and, if necessary, automatically transmit an alarm. This work extends that of a West Point student project in which a liquid-filled sensor pad placed close to the skin captured a signal that was processed by digital filtering to extract the heart and breath signals (Tappert et al. 2001). Recently, we became aware of a new, similar product, the LifeShirt, which is a lightweight shirt with embedded sensors that measure respiratory function, heart rate, posture, and activity level, with optional peripheral devices to measure blood pressure and blood oxygen saturation (VivoMetrics Website 2002).

VoiceXML and InkXML systems

Three projects involved VoiceXML and/or InkXML. The **VoiceXML Application Development Facility** was created to facilitate VoiceXML application development, and several small applications were used to test the

system. The lab provides a gateway to develop voice applications, the interface for developing applications together with a template library, the capability to enable novices to use the code library to learn about and to build advanced applications, and facilitates both novice and expert developers in deploying multiplatform applications. The laboratory hardware consists of a Cisco router for Voice Over IP, the IBM Voice Server (provided by IBM), a Local Area Network (LAN), the Public Switched Telephone Network (PSTN), and a Firewall for the Voice Server. A minimal prototype of the facility is now operational, and upon completion of the facility, the architecture of the VoiceXML studio should permit the registration, development, and deployment of an application. The studio essentially enables the user to load his/her application URL that can be referenced by the IBM Voice Server.

Another team worked on **VoiceXML Applications**. The primary one, a voice-enabled Web-based absentee system (Gallivan et al. 2002), was implemented on the TellMe voice portal (TellMe Voice Portal 2002). It was tested by the class of software engineering students; students who intended to miss a class called the VoiceXML telephone number and were led through an automated dialog to record in a database his/her name, the date and time of the call, the courseID, and the date he/she would miss class. The system provides the instructor and other administrators with a permanent record of absentees that can be accessed and displayed in various forms through a Web interface. A more useful application of such a system would be for the university, or for any reasonably sized organization, to record absences or work time missed by its employees. Rather than calling a secretary, employees would telephone the system to report their absences, and the employer could then access this information through the Web interface; other departments (or systems) needing the information, such as payroll, could also access the database. This team also analyzed a telephone survey application requested by a psychology professor at SUNY New Paltz. This application differs from the usual VoiceXML applications because it requires the server to initiate the telephone calls rather than receive them. In this case, the server places the calls and, when the person answers, the application dialogue conducts the survey and stores the responses in a database. Although this application was not implemented because voice portals like TellMe only allow applications that receive calls, it should now be possible to develop such applications on our new VoiceXML facility.

A third XML system, the **Multimodal Voice/InkXML System**, combined voice and pen input using the standard voice and ink XML formats (Trabelsi et al. 2002a, 2002b). Just as a standard VoiceXML format that facilitates the development of voice applications has been accepted, so the acceptance of a standard InkXML format to facilitate pen applications is anticipated once it is developed. The student team developed a multimodal interface architecture that combines standardized voice and ink formats to facilitate the creation of robust and efficient multimodal systems, particularly for noisy mobile environments. By providing mutual disambiguation of input signals and superior error handling, this architecture should broaden the spectrum of users to the general population, including permanently and temporarily disabled users. Integration of VoiceXML and InkXML provides a standard data format to facilitate Web-based development and content delivery. Diverse applications ranging from complex data entry and text editing applications to Web transactions can be implemented on this system, and a prototype platform and sample dialogues were developed.

Cluster and grid computing system

One team proposed its own project, the **Cluster/Grid Computing System**, and developed a high performance computing facility by co-opting idle and/or underutilized computer workstations to create a virtual computing pool (Clarke et al. 2002). While the idea is not new, it is gaining greater acceptance in academic institutions that possess enough raw computing power to do distributive computing without the need to buy a high-end dedicated computer. During the first semester, the team constructed a cluster computer consisting of five PCs and demonstrated a substantial speed improvement of the cluster over a single PC performing the calculations of a standard ray-tracing problem in computer graphics. For the second semester, the team developed a grid computer for molecular calculations by choosing a computation-intensive molecular simulation package and running tests in the student computer-lab facility that is highly used by students during peak hours. Here, depending on the nature of the simulation with respect to the number of atomic interactions that need to be calculated in real time, it was demonstrated that the more compute nodes attached to the simulation the less intrusive the job was to the user of any workstation in the grid.

Benefits of Real-World Projects

In this section we describe the various benefits of having the computer science students develop real-world systems for real customers. Overall, these projects result in a beneficial outcome for all concerned.

Customers receive valuable systems

The customers benefit from the systems created for them by the students. In some cases, this amounts to a monetary savings, because the customers do not have to outsource the projects. The university itself can greatly benefit by having small projects completed by students. Not only is there a monetary benefit, but also students often do a better job because they are more motivated and may have a better understanding of the university's needs. In other cases, the customers receive systems they might not obtain under ordinary circumstances. Sometimes this is simply because funds for the project were not in the budget. For other situations, such as for the research-oriented systems, there is often no standard procedure for getting the system constructed.

Students develop technology, team, and professional skills

Developing real-world systems for real customers is a stellar real-world learning experience for the students, as individual technologists, as team members, and as maturing professionals in computing. Individually, the students learn technology skills, critical thinking skills, and how to be innovative and creative. They learn the technology skills necessary to develop real-world computer information systems. Through project reviews and team presentations, the students also learn about the various technologies used in other projects, and they especially appreciated the exposure to the projects involving cutting-edge technology and research.

Working in teams, the students learn fair-mindedness, intellectual humility, intellectual integrity, and the ability to work with others to produce useful systems and to take responsibility for them. Because most of the students are employed full time in various areas of computing, they bring their knowledge and expertise to bear in their project work, and by exchanging information, they learn from each other in this student-centered learning environment. The students also benefit from interacting closely with fellow students from diverse cultures and with diverse technological skills.

As maturing professionals, the students learn how to act in the computing field not only as technologists but also as value providers. By working with real customers in developing their project systems and focusing on human-centered computing, the students learn important value skills. This learning paradigm fosters lifelong habits for learning and the application of critical thinking and value skills.

Promotes interdisciplinary collaboration and Pace and local community involvement

These projects involved two communities, the university community and the local community external to the university. Within the university community, many of the projects involved interdisciplinary collaboration with other departments, such as the medical projects with the nursing school, the online survey project with the assessment department, and the involvement with the business school and DoIT. The work with the SUNY New Paltz psychology professor extended our collaboration to the greater university community. The projects also extended into the local community, with the projects involving Northern Westchester Hospital and IBM, to provide the students with off-campus experiences and to foster an extended community for learning and growth. With project customers coming from the local community as well as from the university community, the project work brought to light how knowledge of computing and information technology connects with personal and social responsibility. Interestingly, we found that having projects that dealt with both the Pace community and the local community led to a blurring of the boundary between those communities.

Furthers student and faculty research

Seven of the project systems are related to student and faculty computer science research that is ongoing and expected to continue. Three concern pattern recognition: the Experimental Handwriting System, the User Verification System, and the Rare Coin Grading System; and the first two of these are related to security and personal identification. Three concern XML: the VoiceXML Application Development Facility, the VoiceXML Applications, and the Multimodal Voice/InkXML System. Finally, one concerns wearable computing devices: the Medical Vital Sign Wearable computer. Seven other systems are research-related but are not related to ongoing student-faculty research. Two concern wireless technology: the NIS Wireless Extension and the Emergency Pre-Hospital Care Communication System. Two concern education-related research and were of short-term interest to the faculty customers: the Project Group Assignment System and the Test Item Reliability Analyzer. The two nurse information systems concern handheld computers and the J2ME, J2SE, and J2EE computer languages and, although not representing ongoing research by CSIS students and faculty, are of continued interest to the nursing school as a developing application that, due to its cutting-edge nature, should to be of interest to next year's students as a continuation project. Finally, one deals with supercomputing (short-term student interest): the Cluster/Grid Computing System. Clearly, these real-world projects profoundly foster and further student and faculty research.

Enhances relationships between the university and local technology companies

Some projects drew on the intellectual capital pool of local companies. For example, three projects benefited from collaboration with IBM scientists: the VoiceXML facility team received a copy of the IBM Voice Server from the IBM Speech group, the vital-sign wearable computer team received advice and use of an electronics lab from an IBM scientist, and the Voice/InkXML team interacted with the IBM Pen Technologies group. Furthermore, the IBM Pen Technologies group is interested in an extended collaborative relationship to pursue research in areas of mutual interest and to have students test their handwriting research systems.

Increases national recognition of the university

Many of the benefits indicated above strengthen the university's external image and identity. Students can also present papers on their project work at technical conferences, and those working on cutting-edge technologies can enter competitions. This is already happening – for example, students and faculty have presented papers related to these systems at conferences (see references), and several students won prizes in IBM's VoiceXML contest.

Cost of Real-World Projects

Although far outweighed by the benefits, the cost of using real-world projects is significant. There is an increased load on the instructors to solicit and interact with customers, to set up the projects, to work with the computer support personnel to set up the project development infrastructure, and to monitor the systems development process. There is also significant computing infrastructure required to facilitate the development work.

Setting up projects, forming teams, and assigning projects

The first task is to come up with the projects, and this process is conducted primarily during the summer prior to the beginning of the school year. For this year, eight projects came from those interested in developing systems to further their research (CSIS faculty, sometimes in collaboration with local technology companies, and, in one case, with a DPS student), one came from the instructor's course need (Project Group Assignment System), one came from the students' interest (Cluster/Grid Computing System), and the remaining eight came from contacts with and e-mail solicitations to faculty of CSIS and other departments.

The instructor then works with each of the customers to size and shape the project to be an appropriate systems development experience for the students, and this results in a written project description that is posted on the "projects" page of the course Website. For customers not familiar with computing systems, the instructor often performs most of this task independently. The inducement for the customers' involvement with the students throughout the year is the anticipation of receiving a useful system, although they are warned that the primarily purpose of the projects is to provide the students with a good educational experience and that not all projects are successfully completed.

The next task is to form the student teams and assign each team to a project. This year the students formed their own teams, the teams listed their project preferences, and the instructor assigned teams to projects based almost entirely on their preferences. There are significant problems with this approach, such as the tendency of students to choose group members based solely on friendship and that the assignment of projects can be perceived as unfair by students who did not obtain their first choice. For this reason, one of the projects this year was to develop a group assignment system to automatically assign students to project teams based on compatible project interests, compatible time schedules and geography (work and home location) for group meetings, and the diversity and appropriateness of students' computing and leadership experience; this system will be used next year to automatically group students into teams and to assign projects to the teams.

Systems development process

The development of the computer information systems requires a systematic approach by the students and proceeds through an evolutionary process model consisting of analysis, design, build, and test phases. The first phase requires the students to work closely with their customer to understand what is desired, usually by developing use cases and use scenarios about how the system will be used; to plan the project; and to perform a risk analysis, all culminating in a written requirements document. The second phase involves designing the system using various analysis and design tools, and this work varies greatly depending on the type of system. For example, a database backend requires database design methodologies, whereas a Web interface requires interface and Web design engineering. The third phase is the actual construction of the system and, depending on the project, usually requires the use of one or more computer software languages and/or database implementation. The fourth phase is the testing of the system to ensure that it meets the customer's requirements and that all of its functions operate correctly.

In the software engineering sequence, the students were required to complete a working system prototype with the core components of the required functionality by the end of the first semester. Because the first semester of the course focuses on the traditional systems development methodologies and the second semester on the object-oriented ones, the students reengineered their systems using object-oriented techniques during the second semester to culminate in a robust and complete system. For the pervasive computing course, the systems were completed in one semester, which was possible because of the course's greater emphasis on project work and because most of the students were also in the software engineering course and therefore were familiar with systems development.

Project development infrastructure

Since most of the projects used the client/server architecture, a Pentium II NT development server was set up for the students to develop their systems. The software installed included Cold Fusion, PHP Hypertext Processor, Active Server Pages (ASP), MySQL, Tomcat, and Microsoft Office. This server was independent and separate from the other CSIS servers so that students could not corrupt data or interfere with operations on CSIS production servers. A variety of database-related software was used by the ten client/server systems and the VoiceXML absentee system that also used a backend database and a Web interface, the latter as an alternative to the telephone input and for displaying the results (Table 1). The student teams used three scripting languages (Cold Fusion, PHP, and ASP) to communicate with the databases and two database systems (Microsoft Access and MySQL). Several projects also used Java-related software, such as Java servlets, and Tomcat was installed to handle the processing of that code. An additional server was made available for the VoiceXML facility project and six old Pentium I and II PCs, network cards, and network cables were made available for the cluster computer project.

The project development infrastructure will be enhanced next year to reach a higher level of professionalism in project development. The NT development server will be upgraded and a Unix server added – a Solaris-Oracle configuration. A production server will also be added toward the middle of the spring semester to house the final systems and to give the students practice in moving their system from one environment to another.

Conclusions

In 2001-2, Pace University CSIS students developed real-world computer information systems for actual customers. Developing real-world systems provides an excellent setting for teaching students the technological skills of the discipline, team-related skills, and value skills by following a human-centered development process. It also provides a vehicle for fostering interdisciplinary collaboration, encouraging student involvement in the university and local communities, furthering student and faculty research, enhancing relationships between the university and local technology companies, and increasing national recognition of the university.

References

Baker, W., Evans, A., Jordan, L. & Pethe, S. (2002). User verification system. *Proc. MASPLAS'02.*

Barbosa, C., Pandey, N., Pavlenko, O., Cunning, P. & Pramod, S. (2002). A Web-based genealogy application system. *Proc. MASPLAS'02.*

Bassett, R., Gallivan, P., Gao, X., Heinen, E. & Sakalaspur, A. (2002). Development of an automated coin grader. *Proc. MASPLAS'02.*

Cha, S.-H. & Tappert, C.C. (2002). Automatic detection of handwriting forgery. *Proc. IWFHR-8*, Niagara, Canada, pp. 264-267.

Clarke, C., Marino, L. & Pachigolla, R. (2002). Creating a virtual computing facility. *Proc. MASPLAS'02.*

Denning, P.J. & Dunham, R. (2001). The core of the third-wave professional. *Communications of the ACM*, Vol. 44, No. 11, pp. 21-25.

Dertouzos, M. (2001). *The Unfinished Revolution.* Harper.

Doyle, K., Kroha, S., Palchowdhury, A. & Xu, W. (2002). Project group assignment system. *Proc. MASPLAS'02.*

Gallivan, P., Hong, Q., Jordan, L., Li, E., Mathew, G., Mulyani, Y. & Visokey, P. (2002). VoiceXML absentee system. *Proc. MASPLAS'02.*

ITEMAL (2002). http://www.udel.edu/topics/testscan/itemal.html.

MASPLAS'02 Website (2002). http://csis.pace.edu/csis/masplas/.

Merritt, S.M., Grossman, F., Tappert, C., Bergin, J., Blum, H., Frank, R., Sachs, D., Stix, A. & Varden, S. (2001). The Doctor of Professional Studies in Computing: An Innovative Professional Doctoral Program. *Proc. 19th Annual Info. Systems Ed. Conf.*, ISECON 2001.

Palmer, S., Panchee, N., Sullivan, J., Thabet, K. & Westgard, S. (2002). Migrating an application to Java2 Micro Edition. *Proc. MASPLAS'02.*

Park, H.G., Pastore, Jr., J.M. & Tappert, C.C. (2002). Wireless technologies in pre-hospital communications. *CSIS Technical Report*, No. 176.

Pethe, S., Baker, W., Brown, N., Hennings, R. & Misra, S. (2002). Online course opinion survey system. *Proc. MASPLAS'02.*

Pressman, R.S. (2001). *Software Engineering*, McGraw-Hill.

Redmond, M.A. (2001). A computer program to aid assignment of student project groups. *Proc. 32nd SIGCSE Tech. Sym. CS Ed.*, pp. 134-138.

Tappert, C.C. et al. (2001). Military applications of wearable computers and augmented reality. Chapter 20, pp. 625-647, in *Fundamentals of Wearable Computers and Augmented Reality*, ed. Barfield, W. & Caudell, T., Lawrence Erlbaum.

TellMe Voice Portal (2002). http://www.tellme.com/.

Trabelsi, Z., Tappert, C. & Desai, D. (2002). Integrated VoiceXML and InkXML gateway. *Proc. AVIOS Speech Expo 2002*, San Jose, CA.

Trabelsi, Z., Cha, S.-H., Desai, D. & Tappert, C. (2002). A Voice and Ink XML Multimodal Architecture for Mobile e-Commerce Systems. *2nd ACM Int. Workshop on Mobile Commerce 2002*, Atlanta, GA.

USC CSE Website (2002). http://sunset.usc.edu/index.html.

VivoMetrics Website (2002). http://www.vivometrics.com/cc/.

BEST COPY AVAILABLE

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

NOTICE

Reproduction Basis

[X] This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

[ ] This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").